Neural networks
Lab 7: Hopfield networks (associative memories), Elman networks (time series modeling)

---

1. Hopfield networks and associative memories.

Architecture, functioning and designing. A Hopfield network consists of N fully connected neurons which are both input, functional and output units. It usually process encoded data, i.e. vectors with components in {-1,1}. Therefore the typical activation function is signum (satlins in Matlab).

An associative memory is a system which allows the storage of a set of vectors in $\{-1,1\}^N$ and the retrieval of them starting from some clues (incomplete or noisy versions of the stored vectors). The storage process consists in setting the values of the weights such that all vectors which should be stored become stable fixed points of the network dynamics.

The dynamics of the network, for a given input $X=(x_1,\ldots,x_N)$, can be described by some recurrence relations which expresses the relationship between the output produced by the network at time (t+1) and the output of the network at time t:

$$y_i(t+1) = f\left(\sum_{j=1}^{N} w_{ij} y_j(t)\right), \quad i = \overline{1,N}$$

$$y_i(0) = f(x_i), \quad i = \overline{1,N}$$

The simplest rule to store data in a Hopfield network is the so-called Hebb rule. For a set $\{X^1,\ldots, X^L\}$ of vectors to be stored, the Hebb rule leads to the following values of the weights:

$$w_{ij} = \frac{1}{N}\sum_{l=1}^{L} x_i^l x_j^l, \quad i,j = \overline{1,N}$$

Implementation in Matlab. A Hopfield network can be created in Matlab by using the function newhop(data) where data is a matrix containing on its rows the data to be stored.

The network functioning is simulated using the function sim. There are two variants of calling the function sim:

result= sim(net,M,[],test) or result= sim(net,{M,iterations}, {},test)

where M is the number of test data to be taken from the test matrix (specified as the last parameter). In the first variant the user does not control the number of iterations while in the second case he can do this.

Example 1.

```
% Design of a Hopfield network which stores 4 vectors
vectors=[-1 1 -1 -1 1 -1 -1 1 -1; -1 -1 -1 1 1 1 -1 -1 -1; -1 -1 1 -1 1
-1 1 -1 -1; 1 -1 -1 -1 1 -1 -1 -1 1]';
net=newhop(vectors);
result=sim(net,4,[],vectors);
```

```
disp('Stored vectors:');  disp(vectors);
disp('Fixed points:'); disp(result);

% Dest data
test={[0.1; 0.8; -1; -0.7; 0.5; -1; -0.9; 0.85; -1]};
result=sim(net,{1,5},{},test);
% Network state after each iteration
for i=1:5,
   disp(sprintf('Network state after %d iterations:',i));
   disp(result{i});
end
```

Application 1. Design a Hopfield network which implements an associative memory which stores 4 pattern of size 5*5 and check the retrieval ability of the networks for perturbed versions of the stored patterns.

Hint. See HopfieldNet.ma

2. Elman networks and time series modeling.

An Elman network contains besides the typical units in a feedforward network some so-called contextual units which, for each input data, stores the values produced by the hidden units at the previous step (corresponding to the previous input data). Initially the contextual units have the value 0 and the weights corresponding to the connections between them and the hidden units are trained as all the other weight (e.g. by a BackPropagation algorithm). The weights corresponding to the hidden connections (between the hidden units and the contextual ones) are fixed (equal to 1) – they just allow the transfer of the values from the hidden layer to the input layer. The existence of the contextual units allows the Elman network to model time series.

In Matlab an Elman network can be created by using the function newelm for which the usual parameters are as specified in the following call:

newelm(input, output,[N_1,…,N_K],{f_1,…,f_K}]

where input and output represent the training set, N_1, …, N_K denote the number of units on each functional layer and f_1,…,f_K denote the activation function for each functional layer. The default values for the activation functions are tanh for the hidden units and purelin for the output unit.

Application 2. Design an Elman network to model the series $x_n = 0.75x_{n-1}^2 + 0.2x_{n-2}$

(x_1=x_2=1).

Hint. See ElmanNet.ma (the parameters of the function are: n=number of training data, m=number of test data, K=number of hidden units, epochs=number of training epochs.

Exercise. Analyse:

a. the prediction ability of the network for the following values of the parameters n=20, m=30, K=15, epochs=300
b. the influence of the number of hidden units (K) on the prediction quality
c. the influence of the number of epochs on the prediction quality